

ЛЕКЦИЯ №9. СИСТЕМЫ РАСПРЕДЕЛЕННОЙ ОБРАБОТКИ ДАННЫХ. ПЕРЕХОД К РАСПРЕДЕЛЕННОЙ ОБРАБОТКЕ ДАННЫХ. АРХИТЕКТУРА СОВРЕМЕННЫХ РАСПРЕДЕЛЕННЫХ СУБД. ПРИМЕРЫ.

9.1 Предварительные замечания.....	1
9.2 Принципы функционирования распределенной БД.....	1

9.1 Предварительные замечания.

Системы дистрибутивных баз данных состоят из набора узлов, связанных вместе коммуникационной сетью, в которой:

1. каждый узел обладает своими собственными системами баз данных;
2. узлы работают согласованно, поэтому пользователь может получить доступ к данным на любом узле сети, как будто все данные находятся на его собственном узле.

Из этого следует, что так называемая "распределенная база данных" на самом деле является типом виртуального объекта, части которого физически сохраняются в ряду отдаленных "реальных" баз данных на удаленных узлах (фактически, это логическая единица всех этих реальных баз данных).

Каждый узел обладает своими собственными базами данных, собственными локальными пользователями, собственной СУБД и программным обеспечением для управления транзакциями (включая собственное программное обеспечение для блокирования, регистрации, восстановления и т.д.), а также своим собственным локальным диспетчером передачи данных. В частности, пользователь может на собственном локальном узле выполнять операции с данными так, как будто этот узел вовсе не является частью распределенной системы.

9.2 Принципы функционирования распределенной БД

Теперь, после краткого введения, можно привести формулировку фундаментального принципа распределенной базы данных: для пользователя распределенная система должна выглядеть точно так же, как нераспределенная система.

1. Изложенный фундаментальный принцип приводит к следующему набору правил и целей: Локальная автономия; Независимость от центрального узла; Непрерывное функционирование; Независимость от расположения
2. Независимость от фрагментации;
3. Независимость от репликации;
4. Обработка распределенных запросов;
5. Управление распределенными транзакциями;
6. Независимость от аппаратного обеспечения;
7. Независимость от операционной системы;
8. Независимость от сети;
9. Независимость от СУБД.

9.2.1 Локальная автономия

В распределенной системе узлы следует делать автономными. Локальная автономия означает, что операции на данном узле управляются этим узлом, т.е. функционирование любого узла X не зависит от успешного выполнения некоторых операций на каком-то другом узле Y (в противном случае может возникнуть крайне нежелательная ситуация, а именно: выход из строя узла Y может привести к невозможности исполнения операций на узле X, даже если с узлом X ничего не случилось). Из принципа локальной автономии также следует, что владение и управление данными осуществляется локально вместе с локальным ведением учета.

9.2.2 Независимость от центрального узла

Под локальной автономией подразумевается, что все узлы должны рассматриваться как равные. Следовательно, не должно существовать никакой зависимости и от центрального "основного" узла с некоторым централизованным обслуживанием, например централизованной обработкой запросов, централизованным управлением транзакциями или централизованным присвоением имен.

Зависимость от центрального узла нежелательна по крайней мере по двум причинам. Во-первых, центральный узел может быть "узким" местом всей системы, а во-вторых, более важно то, что система в таком случае становится уязвимой, т.е. при повреждении центрального узла может выйти из строя вся система.

9.2.3 Непрерывное функционирование

Одним из основных преимуществ распределенных систем является то, что они обеспечивают более высокую надежность и доступность.

1. Надежность (вероятность того, что система исправна и работает в любой заданный момент) повышается благодаря работе распределенных систем не по принципу "все или ничего", а в постоянном режиме; т.е. работа системы продолжается, хотя и на более низком уровне, даже в случае неисправности некоторого отдельного компонента, например отдельного узла.

2. Доступность (вероятность того, что система исправна и работает в течение некоторого промежутка времени) повышается частично по той же причине, а частично благодаря возможности репликации данных (подробнее это описывается ниже).

9.2.4 Независимость от расположения

Основная идея независимости от расположения (которая также называется прозрачностью расположения) достаточно проста: пользователям не следует знать, в каком физическом месте хранятся данные, наоборот, с логической точки зрения пользователям следовало бы обеспечить такой режим, при котором создается впечатление, что все данные хранятся на их собственном локальном узле.

9.2.5 Независимость от фрагментации

В системе поддерживается фрагментация данных, если некое хранимое отношение в целях физического хранения можно разделить на части, или "фрагменты". Фрагментация желательна для повышения производительности системы, поскольку данные лучше хранить в том месте, где они наиболее часто используются. При такой организации многие операции будут чисто локальными, что снизит нагрузку на сеть.

Существует два основных типа фрагментации – горизонтальная и вертикальная, которые связаны с реляционными операциями выборки и проекции соответственно. Иначе говоря, фрагментом может быть любое произвольное подчиненное отношение, которое можно вывести из исходного отношения с помощью операций выборки и проекции. При этом следует учесть приведенные ниже допущения.

Предполагается без утраты общности, что все фрагменты данного отношения независимы, т.е. ни один из фрагментов не может быть выведен из других фрагментов либо иметь выборку или проекцию, которая может быть выведена из других фрагментов. (Если есть необходимость сохранить одну и ту же информацию в нескольких разных местах, для этого следует использовать механизм репликации системы.). Проекция не должны допускать потерю информации.

9.2.6 Независимость от репликации

В системе поддерживается независимость от репликации, если заданное хранимое отношение или заданный фрагмент могут быть представлены несколькими различными копиями, или репликами, хранимыми на нескольких различных узлах.

9.2.7 Обработка распределенных запросов.

Вопрос оптимизации более важен для распределенной, нежели для централизованной системы. Основная причина заключается в том, что для выполнения

ЛЕКЦИЯ №9. СИСТЕМЫ РАСПРЕДЕЛЕННОЙ ОБРАБОТКИ ДАННЫХ. ПЕРЕХОД К РАСПРЕДЕЛЕННОЙ ОБРАБОТКЕ ДАННЫХ. АРХИТЕКТУРА СОВРЕМЕННЫХ РАСПРЕДЕЛЕННЫХ СУБД. ПРИМЕРЫ. охватывающего несколько узлов запроса существует довольно много способов перемещения данных по сети. В таком случае чрезвычайно важно найти наиболее эффективную стратегию.

9.2.8 Управление распределенными транзакциями.

Существует два основных аспекта управления обработкой транзакций, а именно: управление восстановлением и управление параллелизмом, каждому из которых в распределенных системах должно уделяться повышенное внимание.

9.2.10 Независимость от аппаратного обеспечения.

Подразумевает возможность работы узлов системы на разном аппаратном обеспечении.

9.2.11 Независимость от операционной системы.

Подразумевает возможность работы узлов системы под управлением различных операционных систем.

9.2.12 Независимость от сети.

Подразумевает возможность работы узлов системы в гетерогенных сетях, с использованием различного сетевого оборудования

9.2.13 Независимость от СУБД.

Эта цель подразумевает использование несколько менее точной формулировки предположения о строгой однородности. В новой форме это предположение означает, что все экземпляры СУБД на различных узлах поддерживают один и тот же интерфейс, хотя они не обязательно должны быть копиями одного и того же программного обеспечения.

9.2.14 Распространение обновления

Как указывалось выше, основной проблемой репликации данных является то, что обновление любого логического объекта должно распространяться на все хранимые копии этого объекта. Трудности возникают из-за того, что некоторый узел, содержащий данный объект, может быть недоступен (например, из-за краха системы или данного узла) именно в момент обновления.

Общая схема устранения этой проблемы (и не единственно возможная в этом случае), называемая схемой первичной копии, будет описана далее.

1. Одна копия каждого реплицируемого объекта называется первичной копией, а все остальные – вторичными.
2. Первичные копии различных объектов находятся на различных узлах (таким образом, эта схема является распределенной).
3. Операции обновления считаются завершенными, если обновлены все первичные копии. В таком случае в некоторый момент времени узел, содержащий такую копию, несет ответственность за распространение операции обновления на вторичные копии.